

UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO (UFERSA)
CENTRO DE CIÊNCIAS EXATAS E NATURAIS (CCEN)
DEPARTAMENTO DE COMPUTAÇÃO
DISCIPLINA: COMPILADORES
ATIVIDADE: TRABALHO PRÁTICO SOBRE ANÁLISE SEMÂNTICA

OBJETIVO: Estender o analisador sintático construído para a verificação parcial da linguagem OWL Manchester Syntax com três tipos de técnicas de análise semântica.

ANÁLISE DA PRECEDÊNCIA DOS OPERADORES

- **Exemplo 1:** as palavras-chave que encabeçam a descrição de uma classe de uma ontologia em OWL devem aparecer em uma determinada ordem:
 - a. Class: *(termo obrigatório)*
 - b. EquivalentTo: *(termo necessário, que pode ou não ser sucedido por SubClassOf:)*
 - c. SubClassOf: *(termo necessário, que pode ou não ser precedido por EquivalentTo:)*
 - d. DisjointClasses: *(termo opcional, mas que sempre precede Individuals:)*
 - e. Individuals: *(termo opcional, mas que sempre sucede DisjointClasses:)*

*/*Trecho da ontologia de pizzas: aqui aparecem quatro cabeçalhos*/*

```
Class: Pizza

SubClassOf:
  hasBase some PizzaBase,
  hasCaloricContent some xsd:integer

DisjointClasses:
  PizzaBase, PizzaTopping

Individuals:
  CustomPizza1,
  CustomPizza2
```

*/*Trecho da ontologia de contratos em IDS: aqui aparecem apenas dois cabeçalhos*/*

```
Class: Evaluated

EquivalentTo:
  BrokerServiceProvider or Connector or CoreParticipant

SubClassOf:
  FunctionalComplex
```

- **Exemplo 2:** axiomas de fechamento só podem aparecer depois de declarações existenciais de propriedades, pois não tem como fechar as imagens das propriedades sem tê-las declarado antes explicitamente.

/*Trecho da ontologia de pizzas: neste caso, o axioma de fechamento para a propriedade hasTopping (que restringe as imagens da propriedade às classes MozzarellaTopping ou TomatoTopping) só pode aparecer depois que as duas triplas existenciais (com o operador some) forem declaradas */

Class: MargheritaPizza

SubClassOf:

NamedPizza,
hasTopping some MozzarellaTopping,
hasTopping some TomatoTopping,
hasTopping only (MozzarellaTopping or TomatoTopping)

DisjointClasses:

AmericanaHotPizza, AmericanaPizza, SohoPizza

Individuals:

MargheritaPizzal,
MargheritaPizza2

/*Esse tipo de declaração estaria errada*/

SubClassOf:

NamedPizza,
hasTopping only (MozzarellaTopping or TomatoTopping),
hasTopping some MozzarellaTopping,
hasTopping some TomatoTopping

VERIFICAÇÃO ESTÁTICA DE TIPOS POR COERÇÃO

- **Exemplo 1:** conforme discutido anteriormente, data properties são caracterizadas por unirem uma classe a um tipo de dado. Esses tipos são normalmente precedidos por outro termo-chave denominado namespace, que é um prefixo de um escopo ou linguagem que define aquele tipo de dado. Por exemplo, o namespace “xsd:” é uma abreviação de “**XML Data Schema Definition**”. Portanto, uma declaração do tipo “xsd: integer” declara um inteiro conforme as definições de tipo nativas de XML. Além disso, é possível também indicar faixas de valores que esses tipos podem assumir. Observe o exemplo a seguir. O tipo inteiro é restrito a valores superiores a 400. Neste caso, é preciso delimitar usar operadores relacionais para delimitar o intervalo, p.ex.: **>=400** ou **==500** ou **<=300**. Ou seja, este é um caso onde é preciso “forçar” quem escreve em OWL a indicar os intervalos dos tipos numéricos, caso se aplique. No caso de não ser delimitado um intervalo, a abrangência do tipo compreende todo o conjunto denotado pelo tipo, p.ex.: o conjunto de todos os números inteiros possíveis. Entretanto, no exemplo abaixo, é preciso delimitar o que diferencia uma pizza hipercalórica de uma hipocalórica.

```
Class: HighCaloriePizza
```

```
EquivalentTo:  
    Pizza  
        and (hasCaloricContent some xsd:integer[>= 400])
```

```
Class: LowCaloriePizza
```

```
EquivalentTo:  
    Pizza  
        and (hasCaloricContent some xsd:integer[< 400])
```

- **Exemplo 2:** um outro exemplo de aplicação da coerção como técnica de análise semântica em compiladores pode ser extraído do trecho de ontologia a seguir. Neste caso, há duas propriedades de tipos de dados: **hasTopping** e **ssn**. Ambas são sucedidas pelo operador **min**, o qual pede um numeral logo a seguir. Note que, logo após o numeral, pode aparecer tanto o nome de uma classe (p.ex.: **PizzaTopping** após **hasTopping**) quando um tipo de dado (**xsd:string**, após a propriedade **ssn**). Em ambos os casos, é importante “forçar” o ontologista a indicar um número entre o operador (que pode ser **min**, **max** ou **exactly**) e a imagem da propriedade (nome de classe ou tipo de dado).

```
Class: InterestingPizza
```

```
EquivalentTo:  
    Pizza  
        and (hasTopping min 3 PizzaTopping)
```

```
Class: Employee
```

```
SubClassOf:  
    Person  
        and (ssn min 1 xsd:string)
```

VERIFICAÇÃO ESTÁTICA DE TIPOS POR SOBRECARGAMENTO

- **Exemplo 1:** a declaração de *data properties* em OWL pede que uma propriedade tenha uma classe como domínio e um tipo de dado como imagem. Observe o exemplo a seguir. A propriedade **hasPhone** é seguida de um quantificador (**some**) e um tipo de dado (**xsd:string**). A verificação do tipo de termo que sucede uma propriedade pode ser usada para classificar a propriedade. Neste caso, pode-se inferir que a propriedade **hasPhone** é do tipo “data property”.

```
Class: Customer
```

```
EquivalentTo:  
    Person  
        and (purchasedPizza some Pizza)  
        and (hasPhone some xsd:string)
```

- **Exemplo 2:** uma propriedade de objeto (*object property*) é aquela que liga uma classe a outra classe. Observe o exemplo a seguir. As propriedades “*emitsReport*” e “*mediates*” são object properties porque são sucedidas por um quantificador (some) e nomes de outras classes (*EvaluationReport* e *EvaluationAct*, respectivamente). Deve ser possível deduzir o tipo das propriedades (emitsReport e mediates) como sendo object property analisando o termo que as sucede.

```
Class: EvaluationFacility

SubClassOf:
    IntermediaryParticipant,
    emitsReport some EvaluationReport,
    mediates some EvaluationAct
```

DESAFIO: o projeto consiste estender o analisador sintático com análise semântica de forma a ajudar um ontologista a: (1) escrever as declarações usando a ordem correta dos operadores de cabeçalho (Class, SubclassOf, EquivalentTo, DisjointClasses, Individuals); (2) escrever corretamente os tipos e seus respectivos intervalos que compõem as data properties; e (3) classificar as propriedades em data properties e object properties, por sobrecarregamento).

OBSERVAÇÕES IMPORTANTES:

- Identificar pelo menos UMA ocorrência de cada caso: **precedência dos operadores** (4 pontos), **coerção** (3 pontos), **sobrecarregamento** (3 pontos).
- 1/3 da turma deverá apresentar o trabalho;
- Para os que irão apresentar, 60% da nota corresponde à implementação, enquanto que 40% equivalem à apresentação;
- Os grupos seguem os mesmos das tarefas anteriores;
- Data de entrega: **23/04/2024, até às 23h59.**

REFERÊNCIAS:

Horridge, M., Drummond, N., Goodwin, J., Rector, A. L., Stevens, R., & Wang, H. (2006, November). The Manchester OWL syntax. In *OWLed* (Vol. 216). Disponível online em: https://ceur-ws.org/Vol-216/submission_9.pdf

Protégé Ontology Editor, disponível para download em: <https://protege.stanford.edu/>

Pizza Ontology, disponível online em: <https://protege.stanford.edu/ontologies/pizza/pizza.owl>

THAIN, Douglas. **Introduction to compilers and language design**. Lulu. com, 2016.